UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/585,680	07/10/2006	Xiaodan Jiang	INT.P036	6538

45512 7590 04/12/2011

LAWRENCE CHO ATTORNEY AT LAW
C/O CPA GLOBAL
P.O. BOX 52050
MINNEAPOLIS, MN 55402

EXAMINER	
MITCHELL, JASON D	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
04/12/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/585,680	JIANG ET AL.	
	Examiner	Art Unit	
	JASON D. MITCHELL	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 January 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 January 2011* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☒ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This action is in response to an amendment filed on 1/31/11.

Claims 1 and 3-20 are pending in this application.

Response to Arguments

Support for the Amendments

In the first par. on pg. 10 the applicants state:

> Support for the amended claims, paragraphs, and replacement sheets of the drawings can be found at paragraphs [0004]-[0091] of the specification, Figures 1-11 in the drawings, and claims 1-20 as originally filed. No new matter has been added.

This statement references the entirety of the disclosure and thus constitutes no more than a bare assertion that no new matter has been given and is given no weight.

Objections to the Specification

The deletion of par. [0074]-[0077] is sufficient to overcome the previous objection to par. [0074] which is consequently withdrawn.

Changes to the Drawings

The amendments to the drawings are accepted.

Claim Rejections – 35 USC § 101

The amendments to claims 13 and 16 are sufficient to overcome the previous 35 USC § 101 rejections of claims 13-20 which are consequently withdrawn.

Claim Objections

The amendments to claims 4, 6, 15, 17 and 19 are sufficient to overcome some but not all objections to the claims which are consequently partially withdrawn.

Claim Rejections – 35 USC § 112 1st par.

In the last par. on pg. 11 the applicants state:

> Applicants respectfully submit that "weight" is a term used in the relevant art to assess nodes and instructions. Applicants refer the Office, for example, to the following publications. "A Modified State Reduction Algorithm for Computing Weight Enumerators for Convolution Codes" by E.K.S. Au and Wai Ho Mow, published in Information Theory, 2005, ISIT 2005, Proceedings, International Symposium. "Chinese Automatic Summarization Based on Thematic Sentence Discovery" by Meng Wang, Chungui Li, and Xiaorong Wang, published in Fuzzy Systems and Knowledge Discovery, 2007, FSKD 2007, Fourth International Conference. "Hardware Efficient LBIST with Complementary Weights" by Liyang Lai Patel, J.H. Rinderknecht, and T. Wu-Teng Cheng, published in Computer Design: VLSI in Computers and Processors, 2005, ICCD 2005; Proceedings, 2005 IEEE International Conference.

The applicants appear to assert that the consistent use of the term "compute weight" was merely a typographical error and that the specification was intended to refer to a "computed weight" as apposed to a particular type of weight. For the purposes of this examination this assertion will be accepted and the corresponding amendments will be entered.

Further, it is noted that none of the cited references[1] appear to deal with relevant art, specifically:

[1] The applicant did not provide copies of these references with this submission. Accordingly the examiner has provided copies with this action.

The Au reference describes a "weight enumerator [that] characterizes the codeword distance distribution and allows the error probability bounds of the code to be conveniently computed" (Abstract). "Codeword distance" and "error probability bounds" would not have been useful in the context of balancing nodes across streams as claimed.

The Wang reference describes a method for "extracting the most relevant sentences from [an] original document to form a summary" (Abstract) in the context of an internet search (see e.g. section 1, 1st par. "information retrieve method"). Generating a summary of a Chinese text document would not have been useful in the context of balancing nodes across streams as claimed.

The Lai reference describes a "logic BIST (built-in self test) scheme with complementary weights" (abstract). Calculating the weight of "test vectors" in order to group them together (section 2, 2nd par.) would not have been useful in the context of balancing nodes across streams as claimed.

The examiner's survey of the relevant arts (e.g. class 718 sub-classes 102-106) has produced a number of distinct and often incompatible meanings (see e.g. US 5,872,972; US 6,587,866; US 6,601,084; US 6,728748). Accordingly, it does not appear that those of ordinary skill in the art would have been sufficiently informed of the applicants' intended meaning, or at least that the applicants have failed to disclose the 'best mode' of calculating the claimed 'weight'.

Regardless, for the purposes of this examination, the examiner takes the applicants statement (cited above) as an acknowledgement that calculating a weight, in the claimed context, was well know in the prior art. In view of the acknowledgement that load (i.e. stream) balancing in general was known in the art (see par. [0003] of the specification as originally filed) and this new acknowledgement that calculating a 'weight' for these streams was also known, it does not appear that any reasonable analysis would have expected that these claims (broadly directed to load balancing as a function of a calculated weight) would distinguish over the prior art. Accordingly, the grounds of rejection presented in this action do not merit a second non-final action.

Claim Rejections – 35 USC § 112 2nd par.

The amendments to claims 8, 17, 18 and 19 are sufficient to overcome some but not all 35 USC 112 2nd rejections. Other issues, as indicated below, remain.

Claim Rejections – 35, USC § 102(b) and 103(a)

"partitioning memory access dependence chains"

In the 2nd par. on pg. 13 the applicants state:

> In contrast, Ha discloses partitioning a dataflow Von Neumann RISC hybrid (DAVRID) graph by cutting remote arcs logically. The DAVRID graph is partitioned using dependence set and dominance set, where the dependence set is a variant of Iannucci's dependence set (see Ha Section 4.1). The DAVRID graph is a graph of nodes that includes addition, subtraction, multiplication, comparison, and other arithmetic operations which are not memory access operations (see Ha Figure 3). Clearly, the DAVRID graph is not a memory access dependence chain (see Ha Figure 3). As such, Ha does not teach or suggest partitioning a memory access dependence chain, as claimed.

The examiner respectfully disagrees. First it is noted that the applicants have failed to explicitly indicate a distinction between the claimed "memory access dependence chain" and the disclosed "DAVRID graph". Accordingly, the arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Further, The claims to not describe any particular structure or functionality of an "memory access dependence chain". The specification (at par. [0036] of the specification as originally filed) states:

> .. a memory access dependence chain is a path (n_1, n2, . . . n_k) in the memory access dependence graph where n_1 has no predecessors and n_k has no successor, and can be computed by traversing the memory access dependence graph.

Ha discloses that the DAVRID graph represent data dependency (pg. 85 col. 1 1st partial par. "Note that arcs in the DAVRID graphs ... denote data ... dependencies between nodes") and memory access instructions (pg. 90, col. 2, 1st full par. "Instructions are classified into ... memory access"). Accordingly, the disclosed partitioning of DAVRID graphs can reasonably be said to anticipate the broadly claimed "partitioning a memory access dependence chain".

"partitioning instructions in code into a plurality of pipeline stages to be executed in parallel"

In the par. bridging pp. 13 and 14 the applicants state:

> In contrast, Ha discloses a thread formation scheme to produce sequential threads from programs written in a lenient parallel language (see Ha Abstract). Thus, not only does Ha not teach or suggest partitioning instructions in code into a plurality of pipeline stages to be executed in parallel, Ha in fact, by nature of producing sequential threads, teaches away from partitioning instructions in code into a plurality of pipeline stages to be executed in parallel.

The examiner respectfully disagrees. The applicants appear to be misreading the reference. Ha's reference to "sequential threads" is describing the internal nature of the threads (i.e. the instructions in the thread are executed sequentially see e.g. pg. 83, col. 1, last full par. "A thread is usually defined by a sequence of instructions"). Those of ordinary skill in the art would have understood Ha to be describing a system in which a number of threads are created to be executed in parallel. For example:

- "Multithreading is attractive in a large-scale parallel system" (Abstract)

- "Parallel architectures with a number of off-the-shelf microprocessors" (pg. 83, col. 1, last par.)
- "Over thread level, we exploit ... massive parallelism" (pg. 84, col. 1, 2nd full par.)

Accordingly, far from teaching away, it should be seen that Ha discloses partitioning instructions in code into a plurality of pipeline stages to be executed in parallel.

"the number of desired upstream nodes is the length of the memory access dependence chain divided by a pipelining degree"

In the first full par. on pg. 15, the applicants state:

> Applicants submit that Blelloch discloses partitioning N selected tasks to p groups. However, N is described by Blelloch as being a number of available tasks which have the highest assigned priority, not a number reflecting all available tasks (see Blelloch column 4, lines 13-17). Thus, Blelloch does not teach of suggest a length of the memory access dependence chain divided by a pipelining degree, as claimed.

The examiner respectfully disagrees. Blelloch, col. 4, lines 18-20 teaches dividing a number of perspective tasks approximately evenly across the available processors. This is what is described by the claim language. More specifically, the claimed "number of desired upstream nodes" is broadly equivalent to Belloch's "N selected tasks" and the claimed "pipelining degree" is broadly equivalent to Belloch's "number of available processors [p]". Thus as indicated in the rejection, in the context of Ha, Belloch teaches the claimed limitation.

Claim Objections

Claim 15 is objected to because of the following informalities:

Claim 15 recites "an upstage stream by assigning a first number of desired upstream nodes to the upstream stage". It appears that the applicants have inadvertently transposed the terms 'stage' and 'stream' in the first part of this limitation. It is believed the claim would more properly read "an upstream stage by assigning ...".

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 6 and 17-20 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claim 6 recites "if a computed weight of the upstream stage exceeds a predetermined value". While the specification, as amended, refers repeatedly to a "computed weight" nowhere do the applicants disclose what a "computed weight" is or, more importantly, how it would be calculated. Further, the term "weight" does not appear to have a single specific meaning in the relevant arts. Accordingly those of ordinary skill in the art would not have been enabled to calculate a "compute weight of

the upstream stage" in accordance with the applicants disclosed embodiments without undue experimentation.

Claims 17-19 make similar reference to a "compute weight" and are rejected accordingly.

Claim 20 depends on claim 19 and is rejected accordingly.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 6 and 19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 6 recites "generating a new number of desired upstream nodes". This language could be understood to describe creating a new set of nodes for the upstream or alternately, calculating a different number of nodes to include in the upstream. Accordingly the language of the claim does not particularly point out or distinctly describe the intended scope of the claim.

Further claim 6 recites "assigning a first new number of desired upstream nodes". It is not clear if this "first new number" is intended to refer to the "new number" generated in the previous limitation or to some other new number.

For the purposes of this examination the claim will be read as calculating a different number of nodes to include in the upstream and redistributing nodes so that the upstream holds this (presumably lower) number of nodes. This interpretation

appears to be consistent with the specification (see e.g. pars [0051]-[0052] of the specification as originally filed). The examiner suggests rewriting the claim in its entirety to more clearly recite the desired scope.

Claim 19 recites the limitations "the length unit", "the assignment unit" and "the close up unit". There is insufficient antecedent basis for these limitations in the claim. Claim 19 currently depends from claim 16 which does not provide sufficient antecedent basis for these limitations. However, it is noted that claim 17 does provide the necessary antecedent basis.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claim 1, 3-4, 7-8 and 10-16 are rejected under 35 U.S.C. 102(b) as being anticipated by "Partitioning a Lenient Parallel Language into Sequential Threads" by Ha, Han and Kim.

Claim 1: Ha discloses a method of compiling code, comprising:

partitioning instructions in the code among a plurality of processors based on memory access latency associated with the instructions (pg. 85, col. 2, Section 4 "DAVRID graphs are first partitioned based on only long latency instructions") by

partitioning memory access dependence chains (pg. 85, col. 2, Section 4.1 2nd par. "We partition the DAVRID graphs by ... using dependence set"; (pg. 84, col. 1, 2nd full par. "Over thread level, we exploit ... massive parallelism").

Claim 3: The rejection of claim 1 is incorporated; further Ha discloses partitioning instructions comprises partitioning a memory access dependence chain into an upstream stage (pg. 85, col. 2, Section 4.1 2nd par. "We partition the DAVRID graphs by ... using dependence set"; because the term 'upstream' is at best loosely defined in the specification, it appears that any first partition will meet this claim limitation).

Claims 4 and 15: The rejections of claims 1 and 13 are incorporated; further Ha discloses partitioning instructions comprises partitioning a memory access dependence chain into an upstream stage by assigning a first number of desired upstream nodes to the upstream stage, and assigning instructions in the code on which the first number of desired upstream nodes are dependent to the upstream stage (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically ... using dependence set developed by Iannucci[4]"; pg. 85 col. 1 1st partial par. "Note that arcs in the DAVRID graphs ... denote data ... dependencies between nodes").

Claim 7: The rejection of claim 3 is incorporated; further Ha discloses partitioning the memory access dependence chain into a downstream stage (pg. 85, col. 2, Section 4.1 2nd par. "We partition the DAVRID graphs by ... using dependence set"; again because

the term 'downstream' is, at best, loosely defined in the specification, it appears that any second or subsequent partition will meet this claim limitation).

Claim 8: The rejection of claim 7 is incorporated; further Ha discloses partitioning the memory access dependence chain into the downstream stage comprises:

assigning a last number of desired downstream nodes to the downstream stage (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically"); and

assigning instructions in the code which are dependent on the last number of desired downstream nodes to the downstream stage (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by ... using dependence set developed by Iannucci[4]"; pg. 85 col. 1 1st partial par. "Note that arcs in the DAVRID graphs ... denote data ... dependencies between nodes").

Claim 10: The rejection of claim 1 is incorporated; further Ha discloses identifying instruction dependence information (pg. 85, col. 2, Section 4.1, 2nd par. "dependence set developed by Iannucci[4]").

Claims 11 and 14: The rejections of claims 1 and 13 are incorporated; further Ha discloses constructing a memory access dependence graph (pg. 85, col. 2, Section 4.1, 2nd par. "dependence set developed by Iannucci[4]").

Claim 12: The rejection of claim 1 is incorporated; further Ha discloses:

constructing a memory access dependence graph (pg. 85, col. 2, Section 4.1, 2nd par. "dependence set developed by Iannucci[4]"); and

identifying a memory access dependence chain from the memory access dependence graph (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by ... using dependence set developed by Iannucci[4]").

Claim 13: Ha discloses an article of manufacture comprising a non-transitory machine accessible medium including sequences of instructions, the sequences of instructions including instructions which when executed cause the machine to perform:

partitioning instructions in code into a plurality of pipeline stages to be executed in parallel by a plurality of processors based on memory access latency associated with the instructions (pg. 85, col. 2, Section 4 "DAVRID graphs are first partitioned based on only long latency instructions").

Claim 16: Ha discloses a code analysis unit implemented on a processor, comprising:

a dependence information unit to identify dependencies between instructions in code (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by ... using dependence set developed by Iannucci[4]"); and

a code partitioning unit to partition instructions in the code into a plurality of pipeline stages to be executed by a plurality of processors based on memory access

latency associated with the instructions (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically").

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 5 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Partitioning a Lenient Parallel Language into Sequential Threads" by Ha, Han and Kim in view of US 5,768,594 to Blelloch et al.

Claim 5: The rejection of claim 4 is incorporated; further Ha does not disclose the number of desired upstream nodes is the length of the memory access dependence chain divided by a pipelining degree.

Blelloch teaches partitioning code such that the number of desired upstream nodes is the length of the memory access dependence chain divided by a pipelining degree (col. 4, lines 19-22 "In step 610, the assignment manager AM1 partitions the N selected tasks to p groups of size approx (N/p) each, where p is the number of available processing elements PE1").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to partition Ha's selected tasks (Ha pg. 85, col. 2, Section 4 "first partitioned based on only long latency instructions"; pg. 85, col. 2, Section 4.1, 2nd par. "using dependence set developed by Iannucci[4]") such that the number of desired upstream nodes is the length of the memory access dependence chain (Ha pg. 85, col. 2, Section 4.1, 2nd par. "using dependence set developed by Iannucci[4]"; Blelloch col. 4, lines 19-22 " the N selected tasks ... (N/p)") divided by a pipelining degree (Blelloch col. 4, lines 19-22 " (N/p) ... where p is the number of available processing elements PE1"). Those of ordinary skill in the art would have been motivated to do so as a known method for partitioning the instructions (Blelloch col. 4, lines 19-22 "partitions the N selected tasks") which would have provided the disclosed load balancing (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing").

Claim 9: The rejection of claim 8 is incorporated; further Ha does not disclose the number of desired downstream nodes is N*(d- 1)/d, where N is a length of the memory access dependence chain, and d is a pipelining degree.

Blelloch teaches partitioning code such that the number of desired downstream nodes is N*(d- 1)/d, where N is a length of the memory access dependence chain, and d is a pipelining degree (col. 4, lines 19-22 "In step 610, the assignment manager AM1 partitions the N selected tasks to p groups of size approx (N/p) each, where p is the number of available processing elements PE1"). Note that in a two processor system

Blelloch's partition sized is as claimed (i.e. N*(2-1)/2 = N/2). Further, in a system with more than two processors the second and subsequent partitions can be considered the 'downstream' partition resulting in the claimed partition size (i.e. N*(d-1)/d = N/p*(p-1), where d=p). This understanding appears to match the disclosed process where the downstream partition is itself recursively partitioned into upstream and downstream partitions.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to partition Ha's selected tasks (Ha pg. 85, col. 2, Section 4 "first partitioned based on only long latency instructions"; pg. 85, col. 2, Section 4.1, 2nd par. "using dependence set developed by Iannucci[4]") such that the number of desired downstream nodes is the length of the memory access dependence chain (Ha pg. 85, col. 2, Section 4.1, 2nd par. "using dependence set developed by Iannucci[4]"; Blelloch col. 4, lines 19-22 " the N selected tasks ... (N/p)") divided by a pipelining degree (Blelloch col. 4, lines 19-22 " (N/p) ... where p is the number of available processing elements PE1") resulting in a downstream partition as claimed. Those of ordinary skill in the art would have been motivated to do so as a known method for partitioning the instructions (Blelloch col. 4, lines 19-22 "partitions the N selected tasks") which would have provided the disclosed load balancing (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing").

Claims 6 and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Partitioning a Lenient Parallel Language into Sequential Threads" by Ha, Han and Kim (Ha) in view of US 6,970,929 to Bae et al. (Bae).

Claim 6: The rejection of Claim 4 is incorporated; further Ha discloses performing load balancing (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing") but does not explicitly disclose that load balancing comprises:

generating a new number of desired upstream nodes if a computed weight of the upstream stage exceeds a predetermined value; and

assigning a first new number of desired upstream nodes to the upstream stage; and

assigning instructions in the code on which the first new number of desired upstream nodes are dependent to the upstream stage.

Bae teaches a load balancing procedure (col. 4, lines 32-37 "the load balancing processor 30") comprising:

generating a new number of desired upstream nodes (col. 4, lines 57-63 "the partitioned regions are reestablished") if a computed weight of the upstream stage exceeds a predetermined value (col. 4, lines 57-63 "a weight ... greater than or equal to a predetermined percentage"); and

assigning a first new number of desired upstream nodes to the upstream stage (col. 4, lines 57-63 "the partitioned regions are reestablished"); and

assigning instructions in the code on which the first new number of desired upstream nodes are dependent to the upstream stage (col. 4, lines 57-63 "the partitioned regions are reestablished").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Ha's load balancing (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing") using Bae's methods (col. 4, lines 57-63 "the partitioned regions are reestablished"). Those of ordinary skill in the art would have been motivated to do so as a known method of providing the broadly disclosed functionality (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing").

Claim 17: Ha discloses the apparatus of Claim 16, wherein the code partition unit comprises:

a length unit to determine a number of nodes from a memory access dependence chain to assign to an upstream stage (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically ... using dependence set developed by Iannucci[4]"; pg. 83, 1st full par. "Iannucci[4] gets partitions by aggregating nodes with the same set of input dependencies");

an assignment unit to assign a first number of desired nodes to the upstream stage (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically ... using dependence set developed by Iannucci[4]"; pg. 83, 1st full

par. "Iannucci[4] gets partitions by aggregating nodes with the same set of input dependencies");

a close up unit to assign instructions in the code for which the first number of desired length of upstream nodes are dependent to the upstream stage (pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically ... using dependence set developed by Iannucci[4]"; pg. 83, 1st full par. "Iannucci[4] gets partitions by aggregating nodes with the same set of input dependencies"); and

performing load balancing on the partitioned streams (pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing").

Ha does not explicitly disclose an evaluation unit to determine whether a computed weight of the upstream stage exceeds a predetermined value.

Bae teaches an evaluation unit to determine whether a computed weight of the upstream stage exceeds a predetermined value (col. 4, lines 57-63 "a weight ... greater than or equal to a predetermined percentage").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Ha's load balancing (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing") using Bae's methods (col. 4, lines 57-63 "the partitioned regions are reestablished"). Those of ordinary skill in the art would have been motivated to do so as a known method of providing the broadly disclosed

functionality (Ha pg. 84, col. 2, 1st par. "The message handling unit ... manages ... load balancing").

Claim 18: Ha and Bae teach the apparatus of Claim 17, wherein the length unit determines a new number of desired length of upstream nodes in response to the evaluation unit determining that the computed weight of the upstream stage exceeds the predetermined value (col. 4, lines 57-63 "the partitioned regions are reestablished").

Claim 19: Ha and Bae teach the apparatus of Claim 16, wherein the length unit determines a number of nodes from the memory access dependence chain to assign to a downstream stage, the assignment unit assigns a first number of desired nodes to the downstream stage, the close up unit assigns instructions in the code on which are dependent on the first number of desired length of down stream nodes (Ha pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by cutting remote arcs logically ... using dependence set developed by Iannucci[4]"; pg. 83, 1st full par. "Iannucci[4] gets partitions by aggregating nodes with the same set of input dependencies"; note that this divides the program into nodes and assigns those nodes to both upstream and downstream stages), and an evaluation unit to determine whether a computed weight of the downstream stage exceeds the predetermined value (Bae col. 4, lines 57-63 "a weight ... greater than or equal to a predetermined percentage").

Claim 20: The rejection of claim 19 is incorporated; further Ha discloses a balancing unit to assign remaining instructions to the upstream stage and the downstream stage in a manner that substantially balances computed weight (pg. 84, col. 2, 1st par. "manages the ... load balancing"; pg. 85, col. 2, Section 4.1, 2nd par. "We partition the DAVRID graphs by ... using dependence set developed by Iannucci[4]").

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JASON D. MITCHELL whose telephone number is (571)272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason D. Mitchell/
Primary Examiner, Art Unit 2193